

Oracle Database Administration Workshop I Test Scenario

실습에 앞서 SQL*Plus 환경설정을 다음과 같이 수정합니다. 이미 설정 되어있으면 확인만 합니다.

Practice 1 Oracle Database Architecture

1. ORACLE 백그라운드 프로세스를 확인합니다.

```
$ ps -ef | grep orcl
```

Q1. 오라클 서버의 인스턴스는 ()와 ()으로 구성된다.

Q2. 5가지 필수 백그라운드 프로세스의 이름은 무엇인가?

Q3. 오라클의 주요 메모리의 이름을 쓰시오.

Q4. 세 가지 오라클 데이터베이스 파일을 쓰시오.

Practice 2 Managing the Database Instance

1. 리스너, 데이터베이스를 시작한 후 EMCTL을 이용하여 dbconsole을 시작합니다. EM(Enterprise Manager)를 호출하고 SYS 유저로 로그인합니다.

```
$ lsnrctl start
```

```
$ sqlplus / as sysdba
```

```
SQL> STARTUP
```

```
SQL> exit
```

```
$ emctl start dbconsole
```

a) Mozilla 아이콘을 눌러 웹 브라우저를 엽니다.

b) EM URL을 입력합니다. URL은 http://hostname:portnumber/em 같은 형식입니다.

(예시) http://localhost.localdomain:1158/em.

c) user name에 sys, password는 oracle을 입력하고 connect as를 SYSDBA로 선택하여 로그인 합니다.

2. spfile을 이용해 pfile을 만들고, 초기화 파라미터를 확인합니다. 그리고 JOB_QUEUE_PROCESSES 파라미터를 300으로 설정합니다.

```
$ cd $ORACLE_HOME/dbs
```

```
$ ls ➔ spfileorcl.ora 파일만 현재 있습니다
```

```
$ sqlplus / as sysdba
```

```
SQL> create pfile from spfile;
```

```
SQL> !ls ➔ initorcl.ora file이 생성되었습니다.
```

```
SQL> show parameter block ➔ 결과 중 db_block_size 파라미터의 값 확인
```

SQL> show parameter processes → processes 파라미터 값 확인

SQL> col name for a25

SQL> select name, value, isbasic FROM v\$parameter
WHERE name LIKE '%db_block%' OR name LIKE 'processes%';

SQL> exit

\$ cat spfileorcl.ora → spfile에 job_queue_processes 파라미터의 값 기록 여부 확인

\$ sqlplus / as sysdba

SQL> ALTER SYSTEM SET job_queue_processes = 700 scope=memory;

SQL> show parameter job

SQL> startup force

SQL> show parameter job → 데이터베이스가 재시작되면 변경내용이 취소되었음을 확인

SQL> ALTER SYSTEM SET job_queue_processes = 700;

SQL> startup force

SQL> show parameter job → 데이터베이스가 재시작되어도 변경내용이 유지됨

SQL> !cat spfileorcl.ora → job_queue_processes의 값이 기록되어 있음을 확인

SQL> ALTER SYSTEM SET processes=200; → 정적 파라미터로 SCOPE=spfile을 지정하지 않아 오류 발생

4. 인스턴스 종료 옵션을 테스트합니다. 이 실습은 SYS와 HR 두 개의 터미널을 실행하여 시간순서대로 작업합니다.

SYS	HR
SQL> shutdown transactional	
SQL> startup	SQL> rollback;
	SQL> conn hr/hr
	SQL> select avg(salary) from employees;
	SQL> update employees Set salary = salary *1.1;
	SQL> select avg(salary) from employees;
SQL> shutdown immediate	
SQL> startup	SQL> select avg(salary) from employees;
	SQL> conn hr/hr
	SQL> select avg(salary) from employees;
	SQL> update employees Set salary = salary *1.1;
	SQL> select avg(salary) from employees;

SQL> shutdown abort SQL> startup	SQL> conn hr/hr SQL> select avg(salary) from employees;
-------------------------------------	--

5. Alert Log 및 Trace 파일을 확인하고 관리해 봅니다.

SQL> show parameter background_dump_dest ➔ 경로 확인

SQL> exit

\$ cd \$ORACLE_BASE/diag/rdbms/orcl/orcl/trace

\$ rm *

\$ sqlplus / as sysdba

SQL> startup force

SQL> ! ls ➔ alert_orcl.log file이 다시 생성됨

SQL> !cat alert_orcl.log

SQL> ALTER SYSTEM SET processes = 200 SCOPE = SPFILE ;

SQL> ! tail alert_orcl.log ➔ 파라미터 변경 작업이 기록되었음을 확인

SQL> ALTER SYSTEM SET processes = 150 SCOPE = SPFILE ;

6. Instance 시작을 단계적으로 실행해 봅니다.

SQL> shutdown abort

SQL> startup nomount

SQL> alter database mount;

SQL> alter database open;

Q1. 데이터베이스 초기화 파라미터 파일의 경로와 이름을 쓰시오.

Q2. 데이터베이스 4가지 종료옵션은 무엇입니까?

Q3. 파라미터를 수정하는 명령문은 무엇이며 SCOPE 옵션에 지정가능한 값은 무엇입니까?

Q4. EM을 사용하기위한 emctl 명령어와 EM의 기본 포트를 쓰시오.

Practice 3 Configuring the Oracle Network Environment

1. 다음 경로로 이동하여 Oracle Net 구성파일을 확인합니다.

\$ \$ORACLE_HOME/network/admin

\$ ls

2. 이름이 LISTENER인 기본리스너에 대한 상태를 확인합니다.

```
$ lsnrctl start
```

```
$ lsnrctl status
```

2. 별도의 터미널에서 Net manager를 실행하고 새로운 리스너를 생성하고 확인합니다.

```
$ netmgr
```

a) Oracle Net Configuration > Local > Listeners 선택 후 좌측 + 기호 클릭

b) 리스너 이름 선택 Dialog Box에 Listener 이름을 "L2"로 입력하며 새롭게 추가한 리스너 이름이 Listeners folder에 보임

c) Listeners folder에서 L3 Listener를 선택한 후 화면 오른쪽 Pull-down menu에서 "Listening Locations"를 선택하고 하단의 "Add Address"를 클릭, 리스너의 상세주소를 입력

(Host는 localhost.localdomain, Protocol은 TCP/IP, Port 는 1523로 지정)

d) Pull-down menu에서 "Database Service"를 선택하고 하단의 "Add Database"를 클릭, 필요한 사항을 입력

- 전역데이터베이스 이름 : 비워둠

- Oracle Home : /u01/app/oracle/product/11.2.0/db_1/ ➔ 이미 설정되어 있음

- SID : orcl ➔ 대소문자 구분되므로 반드시 소문자로 입력

e) 상단 메뉴 > file > Save Network Configuration 선택 ➔ 구성내용 listener.ora 파일에 저장

f) Net Manager를 실행하지 않는 다른 터미널에서 리스너 구성파일이 생성되었는지 확인합니다.

```
$ cd $ORACLE_HOME/network/admin
```

```
$ ls
```

```
$ cat listener.ora
```

g) Listener Control Utility를 실행하고 새로운 리스너 "L1"을 관리해 봅니다.

```
$ lsnrctl
```

```
LSNRCTL> start L3
```

```
LSNRCTL> status L3
```

```
LSNRCTL> stop L3
```

```
LSNRCTL> start L3
```

```
LSNRCTL> exit
```

3. Net Manager를 실행하고 프로파일을 설정하여 서비스 이름 지정방식을 변경해 봅니다.

a) \$ netmgr 실행 후 Local > Profile 선택

b) Select Method에서 HOSTNAME 방식을 제거하고 네트워크 구성을 저장합니다.

c) Net Manager를 실행하지 않은 터미널에서 다음을 실행하여 네트워크 구성파일 sqlnet.ora가 생성되었는지 확인합니다.

```
$ ls $ORACLE_HOME/network/admin
```

```
$ cat $ORACLE_HOME/network/admin/sqlnet.ora → NAMES.DIRECTORY_PATH의 값 확인
```

4. Net Manager를 실행하여 orcl 데이터베이스에 접속할 수 있는 새로운 서비스 이름을 생성합니다.

a) Local > Service Naming 선택 후 "+"를 클릭

b) Net Service Name에 mydb를 입력하고 "Next"를 클릭

c) 사용 프로토콜(TCP/IP)과 호스트명(localhost), 그리고 포트번호(1521)를 입력

→ "LISTENER" 리스너 정보와 매치

d) Service Name에 orcl 입력 후 Next → 선택하지 않은 옵션은 기본값 그대로 두고 진행

e) Test 클릭 후 성공여부 확인

→ Test 실패 시 Net Manager 하단의 Change Login 클릭 후 Username과 Password를 HR의 정보로 변경한 후 다시 Net Manager 하단의 Test 클릭하여 확인

f) Test 화면을 Close 클릭하여 종료한 후 finish

g) 네트워크 구성을 저장한 후 Net Manager 종료

h) Net Manager를 실행하지 않은 별도의 터미널에서 tnsnames.ora 파일이 생성되었는지 확인한 후 내용을 봅니다.

```
$ cat $ORACLE_HOME/network/admin/tnsnames.ora
```

5. mydb라는 서비스 이름으로 데이터베이스에 접속해 봅니다.

```
$ sqlplus hr/hr@mydb
```

6. 새로운 터미널에서 "LISTENER" 리스너를 중지한 후 기존 세션의 상태를 알아보고 mydb 서비스 이름으로 데이터베이스에 접속해 봅니다.

a) 기존의 터미널에서 다음을 실행

```
$ lsnrctl stop
```

HR Session

```
SQL> SELECT * FROM departments; ⇒ 기존 접속은 유지
```

b) 새로운 터미널 실행

```
$ sqlplus hr/hr@mydb → 리스너 중지로 접속 실패
```

```
$ sqlplus hr/hr → 로컬접속이므로 리스너 없이 접속 성공
```

- Q1. 기본 리스너의 이름과 사용중인 프로토콜 및 포트번호는 무엇입니까?
- Q2. 서버에서 사용자 접속을 처리하기 위해 필요한 리스너 구성파일의 경로와 이름을 쓰시오.
- Q3. 리스너를 시작하고 종료할 때 필요한 유틸리티 이름과 명령어를 쓰시오.
- Q4. 데이터베이스 사용자가 로컬네임 방식으로 서버에 접속하기 위해 필요한 파일은 무엇입니까?

Practice 4 Managing Database Storage Structures

1. 현재 데이터베이스의 테이블스페이스와 데이터파일, 그리고 HR 테이블이 저장된 테이블스페이스에 대하여 알아봅니다.

```
SQL> conn / as sysdba
```

```
SQL> SELECT tablespace_name, status, contents, extent_management FROM dba_tablespaces;
```

```
SQL> SELECT tablespace_name, file_name, bytes/1024/1024 AS filesize FROM dba_data_files;
```

```
SQL> SELECT table_name, tablespace_name FROM dba_tables
```

```
WHERE owner='HR';
```

2. 새로운 테이블스페이스 INSA를 생성하고 HR이 INSA 테이블스페이스에 테이블을 저장하도록 합니다.

```
SQL> CREATE TABLESPACE insa
```

```
DATAFILE '/u01/app/oracle/oradata/orcl/insa01.dbf' SIZE 10M;
```

```
SQL> SELECT tablespace_name, status, contents, extent_management FROM dba_tablespaces;
```

```
SQL> SELECT tablespace_name, file_name, bytes/1024/1024 AS filesize FROM dba_data_files;
```

```
SQL> conn hr/hr
```

```
SQL> CREATE TABLE emp4
```

```
TABLESPACE insa
```

```
AS SELECT * FROM employees;
```

```
SQL> CREATE TABLE dept4
```

```
(id number(6),
```

```
name varchar2(20))
```

```
TABLESPACE insa;
```

```
SQL> SELECT tablespace_name, table_name FROM user_tables;
```

3. INSA 테이블스페이스를 읽기전용(read only) 상태로 변경한 후 HR의 테이블을 조회해 봅니다.

```
SQL> conn / as sysdba
```

```
SQL> alter tablespace insa read only;
```

SQL> conn hr/hr

SQL> select count(*) from emp4

SQL> update emp4

Set salary = salary *1.1;

SQL> update employees

Set salary = salary *1.1; ➔ USERS 테이블스페이스에 저장되어 있으므로 실행가능

SQL> rollback;

SQL> conn / as sysdba

SQL> ALTER TABLESPACE insa READ WRITE;

4. INSA 테이블스페이스를 OFFLINE 상태로 변경한 후 HR의 테이블을 조회해 봅니다.

SQL> ALTER TABLESPACE insa OFFLINE;

SQL> conn hr/hr

SQL> select count(*) from emp4; ➔ 조회불가

SQL> select count(*) from employees; ➔ USERS 테이블스페이스에 저장되어 있으므로 실행가능

SQL> conn / as sysdba

SQL> ALTER TABLESPACE insa ONLINE

5. INSA 테이블스페이스에 insa02.dbf 파일을 추가하거나 기존 insa01.dbf 파일 사이즈를 변경하여 공간을 증가시킵니다.

SQL> ALTER TABLESPACE insa

ADD DATAFILE '/u01/app/oracle/oradata/orcl/insa02.dbf' SIZE 10M;

SQL> ALTER DATABASE DATAFILE '/u01/app/oracle/oradata/orcl/insa01.dbf' RESIZE 20M;

SQL> SELECT tablespace_name, file_name, bytes/1024/1024 AS filesize FROM dba_data_files;

SQL> SELECT tablespace_name, COUNT(*) AS datafiles, SUM(bytes)/1024/1024 AS filesize

FROM dba_data_files

GROUP BY tablespace_name;

6. INSA 테이블스페이스를 삭제한 후 HR의 emp4 테이블의 유지여부에 대하여 조사해 봅니다.

SQL> DROP TABLESPACE insa INCLUDING CONTENTS AND DATAFILES;

SQL> select count(*) from hr.sawon;

7. EM을 사용하여 USERS 테이블스페이스에 대한 정보를 확인합니다.

a) Server > Storage > Tablespaces

b) USERS 테이블스페이스의 이름을 누릅니다.

c) Warning 임계값을 확인합니다.

d) Actions drop-down list 에서 Show Tablespace Contents를 선택한 다음 Go를 누릅니다.

e) Show Tablespace Contents 페이지가 나타납니다.

f) 페이지 아래쪽으로 이동한 다음 Extent Map 레이블 왼쪽에 있는 더하기(+) 아이콘을 누릅니다.

g) 몇 초 후에 Extent Map이 나타납니다. Map에서 분홍색은 테이블스페이스 헤더입니다.

h) 다시 페이지 맨 위로 이동하여 세그먼트의 이름을 선택하여 그 위치를 확인합니다.

8. EM을 사용하여 크기가 5MB인 INVENTORY 라는 새 테이블스페이스를 생성합니다.

a) EM에서 Server > Storage > Tablespaces

b) Create를 누릅니다.

c) 테이블스페이스 이름으로 INVENTORY를 입력하고 다음을 확인합니다.

- Extent Management : Locally Managed

- Type : Permanent

- Status : Read Write ※ Use Bigfile Tablespace는 선택하지 않습니다.

d) Datafile 영역에서 Add를 누릅니다.

e) Add Datafile 페이지에서 File Name에 inventory01.dbf, File Size에 5MB를 입력하고 Continue를 누릅니다.

f) Storage 탭을 누르고 Extent Allocation 및 Segment Space Management가 Automatic이고 logging이 활성화되었는지 확인합니다.

g) General 탭을 누릅니다.

h) Show SQL을 눌러 실행할 SQL을 확인한 다음 Return을 누릅니다.

i) OK를 누르면 성공 메시지가 나타납니다.

9. EM에서 INVENTORY 테이블스페이스에 대해 5MB 대신 50MB 공간을 정의합니다.

a) Server > Storage > Tablespaces

b) INVENTORY 테이블스페이스를 선택한 다음 Edit를 누릅니다.

c) Datafiles영역에서 Edit를 누릅니다.

d) File Size를 50MB로 변경한 후 Continue를 눌러 General 탭 페이지로 돌아갑니다.

e) Show SQL을 눌러 실행할 SQL을 확인한 후 Return을 누릅니다.

f) Apply를 누릅니다.

Q1. 테이블스페이스 종류를 CONSTANTS에 따라 세 가지로 분류해 보시오.

Q2. 테이블스페이스의 상태에는 어떤 것이 있습니까?

Q3. 테이블스페이스를 삭제하면 저장된 테이블이 어떻게 됩니까?

Q4. 테이블스페이스의 공간을 추가하는 두 가지 방법을 쓰시오.

Practice 5 User Security

1. 관리자로 접속하여 DEMO 사용자가 있으면 삭제 후 DEMO 사용자를 새로 생성한 후 DBA_USERS에서 그 정보를 확인합니다.

```
$ sqlplus / as sysdba
```

```
SQL> drop user demo cascade;
```

```
SQL> create user demo identified by demo;
```

```
SQL> col username for a15
```

```
SQL> col default_tablespace for a15
```

```
SQL> select username, default_tablespace, profile from dba_users
```

```
where username='DEMO';
```

2. DEMO 사용자로 접속을 시도합니다.

```
SQL> conn demo/demo ➔ CREATE SESSION 권한 부족으로 실패
```

3. 관리자로 다시 접속하여 두 가지 권한을 DEMO 사용자에게 부여한 후 접속해 봅니다. 그리고나서 DEMO 사용자의 CLIENT 테이블을 생성합니다.

```
SQL> conn / as sysdba
```

```
SQL> grant create session, create table to demo;
```

```
SQL> conn demo/demo
```

```
SQL> select * from session_privs; ➔ 현재 세션의 권한 확인
```

```
SQL> create table client
```

```
(id number(3),
```

```
name varchar2(20));
```

4. CLIENT 테이블에 데이터 입력을 시도해 봅니다.

```
SQL> insert into client
```

```
values(111, 'Jason'); ➔ 저장 공간을 할당 받지 못하여 실패
```

5. CLIENT 테이블에 데이터를 입력할 수 있도록 DEMO 사용자에게 테이블스페이스의 할당량을 부여합니다.

```
SQL>conn / as sysdba
```

```
SQL> alter user demo quota 5m on users;
```

```
SQL>conn demo/demo
```

```
SQL> insert into client
```

```
values(111, 'Jason');
```

```
SQL> commit;
```

6. DEMO 사용자가 사용할 새로운 테이블스페이스 DEMOTBS를 관리자가 생성한 후, DEMO 사용자의 기본 테이블스페이스를 DEMOTBS로 변경합니다.

```
SQL> conn / as sysdba
```

```
SQL> create tablespace demotbs
```

```
datafile '/u01/app/oracle/oradata/orcl/demotbs01.dbf' size 10m;
```

```
SQL> ALTER USER demo DEFAULT TABLESPACE demotbs;
```

7. DEMO사용자가 새로운 테이블 생성 후 테이블들이 저장된 테이블스페이스를 확인합니다.

```
SQL> conn demo/demo
```

```
SQL>create table account
```

```
(acc_no number(4),
```

```
client_name varchar2(20));
```

```
SQL>select table_name, tablespace_name from user_tables;
```

8. 관리자가 DEMO사용자에게 부여할 demo_r1이라는 이름의 롤을 만들고 권한을 부여한 후 그 롤을 DEMO에게 부여합니다.

```
SQL> create role demo_r1;
```

```
SQL> grant create view, create synonym, create sequence to demo_r1;
```

```
SQL> grant demo_r1 to demo;
```

```
SQL> conn demo/demo
```

```
SQL> select * from session_privs;
```

9. DEMO 사용자가 HR의 EMPLOYEES 테이블 쿼리를 시도해 봅니다.

```
SQL> select * from hr.employees;
```

10. HR사용자가 DEMO 사용자에게 EMPLOYEES 테이블을 쿼리할 수 있는 객체권한을 부여한 후 다시 한번 DEMO사용자가 HR.EMPLOYEES에 대한 액세스를 시도해 봅니다.

```
SQL> conn hr/hr
```

```
SQL> grant select on hr.employees to demo;
```

```
SQL> conn demo/demo
```

```
SQL> select employee_id, last_name from hr.employees;
```

```
SQL> update hr.employees
```

```
2 set salary=salary*1.1; ➔ UPDATE 권한부족
```

11. 다음과 같은 설정으로 DEMO_PROF Profile을 생성하여 할당 받은 사용자에게 다음 사항이 적용되었는지 확인하시오.
RESOURCE_LIMIT 파라미터를 TRUE로 변경하여 자원관련 제한을 활성화 합니다.

```
SQL> conn / as sysdba
```

```
SQL> show parameter resource_limit
```

```
SQL> alter system set resource_limit=true;
```

```
SQL> create profile demo_prof LIMIT
```

```
failed_login_attempts 2
```

```
password_life_time 50
```

```
password_grace_time 10
```

```
sessions_per_user 2
```

```
connect_time 5
```

```
idle_time 2
```

```
password_reuse_max 3;
```

```
SQL> select resource_name, limit, resource_type from dba_profiles
```

```
where profile='DEMO_PROF';
```

```
SQL> alter user demo profile demo_prof;
```

```
SQL> select username, profile from dba_users;
```

12. 터미널을 3개 실행하고 각각 DEMO 사용자로 로그인을 시도해 봅니다. 그리고 그 중 접속에 성공한 한 세션은 IDLE_TIME의 제한을 적용 받기 위해 2분이상 아무 작업도 하지 않은 다음 SQL 명령을 실행해 봅니다.

```
SQL> SELECT * FROM tab;
```

13. DEMO_PROF 프로파일을 삭제한 후 DBA_USERS의 정보를 조회합니다.

```
SQL> conn / as sysdba
```

```
SQL> drop profile demo_prof cascade;
```

```
SQL> select username, profile from dba_users;
```

14. 실습내용을 정리합니다.

```
SQL>conn / as sysdba
```

SQL> drop user demo cascade;

- Q1. 데이터베이스 권한의 2가지 종류는 무엇입니까?
- Q2. 권한을 묶어서 이름을 부여하여 권한관리를 용이하게 한 것을 무엇이라고 하는가?
- Q3. 사용자에게 자원과 암호를 제한하기 위해 사용하는 것을 무엇이라고 하는가?
- Q4. 자원제한을 활성화 하기 위한 파라미터의 이름과 설정값은?
- Q5. 사용자에게 테이블스페이스의 공간을 할당하는 것을 무엇이라고 하는가?

Practice 6 Managing Data Concurrency

1. 터미널을 2개 실행하고 다음과 같이 커밋 되지 않은 갱신을 수행합니다. 시간 순서대로 실습을 실행합니다.

Terminal 1(HR)	Terminal 2(HR)
SQL> conn hr/hr SQL> update hr.employees set salary = salary *1.1;	SQL> conn hr/hr SQL> update employees Set salary = salary *1.05 Where employee_id =101;

2. 세 번째 터미널을 실행하고 관리자 SYS가 문제를 모니터링 합니다.

Terminal 3(SYS)
\$ sqlplus / as sysdba col object_name for a20 col oracle_username for a15 col username for a10 col event for a40 col owner for a15 col name for a20 conn / as sysdba SQL> SELECT session_id, object_id, object_name, oracle_username FROM v\$locked_object JOIN dba_objects USING (object_id); SQL> SELECT session_id, owner, name, mode_held FROM dba_dml_locks; SQL> SELECT sid, serial#, username, blocking_session, event FROM v\$session WHERE username = 'HR' ; ⇒ sid, serial# 열 값 메모 (ex)139,34

3. EM의 Performance 페이지의 Blocking Sessions 링크를 사용하여 어느 세션이 lock 충돌을 유발하고 있는지 알아냅니다.

a) EM에서 Performance 페이지를 누릅니다.

b) Performance 페이지를 처음 호출하면 Software License Agreement가 표시됩니다. [A]를 눌러 동의하고 계속 진행 합니다.

c) Additional Monitoring Links 영역에서 Blocking Sessions를 누릅니다.

4. 차단하는 세션을 종료함으로써 충돌을 해결합니다.

SQL> ALTER SYSTEM KILL SESSION '139,34' ; ➔ 실습 2에서 확인

5. 진행하던 트랜잭션은 모두 rollback 하고 열린 SQL 세션을 모두 종료합니다.

6. Deadlock이 발생하는 경우에 대하여 알아봅니다.

a) 두 개의 터미널에서 각각 HR로 접속하여 다음과 같이 UPDATE를 수행합니다.

HR1	HR2
① SQL> update employees Set salary = salary*1.1 Where employee_id =100; ③SQL> update employees Set salary = salary*1.1 Where employee_id =101; ➔ ORA-00060 : Deadlock detected ...	②SQL> update employees Set salary = salary*1.1 Where employee_id =101; ④SQL> update employees Set salary = salary*1.1 Where employee_id =100; ➔ Waiting

c) 모든 트랜잭션을 ROLLBACK 처리하고 세션을 종료하여 실습을 정리합니다.

Q1. 데이터베이스에서 LOCK 문제가 발생하는 원인을 쓰시오.

Q2. Dead Lock의 발생원인은 무엇입니까?

Practice 7 Managing Undo Data

1. orcl 데이터베이스의 UNDO 설정에 관하여 알아봅니다.

SQL> conn / as sysdba

SQL> show parameter undo

2. 모든 Undo Tablespace 및 Undo Segment들에 관하여 조사해 보고 UNDOTBS1 테이블스페이스에 테이블 생성을 시도해 봅니다.

SQL> select tablespace_name, status, contents

From dba_tablespaces;

SQL> select tablespace_name, segment_name, status

From dba_rollback_segs;

SQL> CREATE TABLE t10

(ID NUMBER)

TABSPACE undotbs1; ⇒ ERROR~!

3. 두 개의 터미널을 더 실행하고 각각 HR로 접속하여 UPDATE를 실행합니다.

HR1	HR2
SQL> update employees Set salary = salary*1.1 Where employee_id =100;	SQL> update employees Set salary = salary*1.1 Where employee_id =101;

➔ Update 중인 두 개의 세션을 그대로 둡니다.

4. 새로운 Undo tablespace UNDOTBS2를 생성하고 모든 Undo Tablespace 및 Undo Segment들에 관하여 조사해 봅니다.

SQL> create undo tablespace undotbs2

Datafile '/u01/app/oracle/oradata/orcl/undotbs2.dbf' size 10m;

SQL> select tablespace_name, status, contents From dba_tablespaces;

SQL> select tablespace_name, segment_name, status From dba_rollback_segs;

5. Undo Tablespace를 UNDOTBS2로 전환한 후 다시 한번 모든 Undo Tablespace 및 Undo Segment들에 관하여 조사해 봅니다.

SQL> alter system set undo_tablespace=undotbs2;

SQL> select tablespace_name, segment_name, status from dba_rollback_segs;

6. 두 개의 활성 트랜잭션을 모두 Rollback 한 후 관리자의 세션에서 Undo Segment의 상태를 다시 한번 확인합니다.

HR1	HR2
SQL> ROLLBACK;	SQL> ROLLBACK;

SQL> select tablespace_name, segment_name, status From dba_rollback_segs;

7. UNDO 테이블스페이스의 공간이 부족한 경우 그 현상과 해결방법을 알아봅니다. 다음과 같이 BIGEMP 테이블을 생성하고 UPDATE를 수행합니다.

a) 다음과 같은 PL/SQL을 실행하여 60만건 이상의 데이터가 저장되는 BIGEMP 테이블을 생성하시오.

SQL> DROP TABLE bigemp purge;

SQL> CREATE TABLE bigemp AS SELECT * FROM employees;

SQL> ALTER TABLE bigemp MODIFY (employee_id NUMBER);

SQL> DECLARE

n NUMBER;

BEGIN

```

FOR n IN 1..14

LOOP

INSERT INTO bigemp SELECT * FROM bigemp;

END LOOP;

COMMIT;

END;

/

```

```
SQL> SELECT COUNT(*) FROM bigemp;
```

b) 많은 양의 Undo data 생성으로 인해 Undo Tablespace 공간부족으로 트랜잭션이 실패하는 상황을 모니터링합니다.

```
SQL> UPDATE bigemp
```

```
SET employee_id =rownum; ➔ 사원번호를 1번부터 순차번호로 변경 시도
```

8. EM을 사용하여 UNDO 테이블스페이스 사용을 모니터 합니다.

Server > Tablespaces 에서 UNDOTBS1과 UNDOTBS2의 Allocated Space Use(%)의 상태를 모니터 합니다.

9. Undo Advisor를 사용하여 최근 1일간의 분석을 기반으로 필요한 언두 공간의 크기를 계산합니다.

a) EM에서 Server > Related Links > Advisor Central을 선택합니다.

b) Automatic Undo Management를 누릅니다.

c) Desired Undo Retention의 Specified manually to allow for longer duration queries or flashback Duration 필드에 2를 입력한 다음 drop-down list에서 days를 선택합니다. Run Analysis 를 누르면 분석결과가 표시됩니다.

d) Analysis Time Period drop-down list에서 Last One Day를 선택합니다. Run Analysis 를 누르면 분석결과가 표시됩니다.

10. 요구되는 사이즈로 언두 테이블스페이스의 크기를 조정하거나 크기를 1GB로 설정합니다.

a) Edit Undo Tablespace를 누릅니다.

b) 기존 데이터 파일의 크기를 늘리기 위하여 해당 데이터파일을 선택한 다음 Edit를 누릅니다.

c) 데이터파일의 사이즈를 변경한 다음 Apply를 눌러 변경 사항을 적용합니다.

11. 실습 내용을 정리합니다.

```
SQL> conn / as sysdba
```

```
SQL> alter system set undo_tablespace = undotbs1;
```

```
SQL> UPDATE hr.bigemp
```

```
SET employee_id =rownum;
```

```
SQL> commit;
```

SQL> drop tablespace undotbs2 including contents and Datafiles;

Q1. UNDO 관련 파라미터는 무엇이 있으며, 그 값은 어떻게 설정되어 있습니까?

Q2. UNDO_TABLESPACE에 지정되지 않은 UNO 테이블스페이스에 속하는 Undo Segment들의 상태는 무엇입니까?

Q3. 테이블의 DML 도중 UNDO 테이블스페이스의 용량이 부족할 경우 어떻게 되며, 문제를 해결하는 방법은 무엇입니까?

Practice 8 Implementing Oracle Database Auditing

1. 표준데이터베이스 감사(audit)로 데이터베이스 작업을 모니터 하려고 합니다. 관련 파라미터를 확인하고 데이터베이스 기본 감사옵션을 확인합니다.

SQL> conn / as sysdba

SQL> show parameter audit_trail

SQL> SELECT privilege, success, failure FROM dba_priv_audit_opts;

SQL> SELECT audit_option, success, failure FROM dba_stmt_audit_opts;

SQL> SELECT owner, object_name, sel, ins, upd, del FROM dba_obj_audit_opts;

2. EM에서 HR.JOBS를 Audit Objects로 선택하고 DELETE, INSERT 및 UPDATE를 Selected Statements로 선택합니다. 감사(audit) 정보는 세션 별로 수집합니다.

a) EM 에서 Server > Security > Audit Settings를 선택합니다.

b) Audited Objects 탭을 누른 다음 Add 버튼을 누릅니다.

c) Add Audited Object 페이지에서 Object Type이 Table인지 확인하고 Table필드에 HR.JOBS를 손전등 아이콘을 사용하여 검색합니다.)

d) DELETE, INSERT, UPDATE를 각각 두 번씩 눌러 Selected Statements 영역으로 이동합니다.

e) Show SQL을 눌러 명령문을 검토하고 Return을 누릅니다.

f) OK를 눌러 감사를 활성화 합니다.

3. 새로운 DB계정 AUDIT_USER 유저를 생성하여 이 유저로 여러 가지 작업을 수행하여 감사레코드 생성 후 유저를 삭제합니다.

\$ sqlplus / as sysdba

SQL> TRUNCATE TABLE aud\$; ➔기존 감사레코드의 삭제

SQL> create user audit_user identified by audit_user;

SQL> grant connect, resource to audit_user;

SQL> grant all on hr.jobs to audit_user;

SQL> connect audit_user/audit_user

SQL> select * from jobs

where job_id LIKE '%MGR';

SQL> SELECT job_id, job_title FROM jobs

where job_id LIKE '%MGR';

SQL> update hr.jobs

set max_salary = max_salary * 10;

SQL> ROLLBACK;

SQL> select * from hr.jobs;

SQL> connect hr/hr

SQL> update hr.jobs

set max_salary = max_salary/10;

SQL> commit;

SQL> select * from hr.jobs;

4. EM 및 SQL*Plus에서 감사된 객체를 검토합니다.

a) Server > Security > Audit Settings 선택

b) 페이지 오른쪽에 있는 Audit Trail 영역에서 Audited Objects를 누릅니다.

c) Audited Objects페이지에서 수집된 정보를 검토합니다.

d) 감사 트레일은 SQL*Plus에서도 다음과 같이 검토할 수 있습니다.

SQL> conn / as sysdba

SQL> col username for a10

SQL> col obj_name for a10

SQL> col owner for a10

SQL> SELECT username, timestamp, owner, obj_name, action_name FROM dba_audit_trail;

5. HRJOBS에 대한 감사 옵션을 취소하고, 감사 레코드를 삭제하여 감사 트레일을 관리합니다. 실습에 사용한 계정을 삭제합니다.

a) Audit Settings 페이지에서 Audited Objects 탭을 누릅니다.

b) Schema에 HR 을 입력한 다음 Search를 누릅니다.

c) 세 개의 행을 모두 선택한 다음 Remove를 누릅니다.

d) Confirmation 페이지에서 yes를 눌러 제거를 확인합니다.

e) 감사 트레일을 SQL*Plus에서 제거한 후 확인합니다.

```
SQL> conn / as sysdba
```

```
SQL> TRUNCATE TABLE aud$;
```

```
SQL> SELECT username, timestamp, owner, obj_name, action_name FROM dba_audit_trail;
```

```
SQL> drop user audit_user cascade;
```

```
SQL> exit;
```

6. 두 개의 별도의 터미널에서 SQL*Plus를 실행하여 Fine-Grained Auditing을 테스트합니다.

a) HR 사용자에게 DBMS_FGA 패키지 사용권한을 부여합니다.

```
$ sqlplus / as sysdba
```

```
SQL> grant execute on dbms_fga to hr;
```

b) 실습에 사용할 HR 소유의 테이블을 생성 합니다.

```
$ sqlplus hr/hr
```

```
SQL> CREATE TABLE x_emp AS
```

```
SELECT employee_id id, last_name name, job_id job, salary FROM employees;
```

c) HR 세션에서 FGA_XEMP 라는 이름의 정책을 생성합니다.

```
SQL> begin
```

```
dbms_fga.add_policy(
```

```
object_name => 'X_EMP',
```

```
policy_name => 'FGA_XEMP',
```

```
audit_column => 'ID, NAME',
```

```
statement_types => 'SELECT, UPDATE',
```

```
audit_column_opts => DBMS_FGA.ALL_COLUMNS);
```

```
end;
```

```
/
```

7. HR에서 SQL 작업 후 Audit Trail을 확인하며 지정된 작업만 감사기록이 남는 것을 확인합니다.

```
SQL> SELECT name FROM x_emp
```

```
WHERE name ='Fay';
```

```
SQL> SELECT id, name FROM x_emp
```

```
WHERE name='Fay';
```

```
SQL> SELECT name, id, job FROM x_emp
```

```
WHERE name='Abel';
```

```
SQL> update x_emp
```

```
set name='Oracle'
```

```
Where id=103;
```

```
SQL> rollback;
```

```
SQL> conn / as sysdba
```

```
SQL> col policy_name for a15
```

```
SQL> col sql_bind for a20
```

```
SQL> col sql_text for a40
```

```
SQL> SELECT policy_name, scn, sql_text, sql_bind FROM dba_fga_audit_trail;
```

```
SQL> SELECT policy_name, scn, sql_text, sql_bind
```

```
FROM dba_fga_audit_trail;
```

```
SQL> SELECT policy_name, scn, sql_text, sql_bind
```

```
FROM dba_fga_audit_trail;
```

8. Audit Trail 관리 및 정책 삭제를 실습해 봅니다.

```
SQL> conn / as sysdba
```

```
SQL> truncate table fga_log$;
```

```
SQL> SELECT policy_name, scn, sql_text, sql_bind
```

```
FROM dba_fga_audit_trail;
```

```
SQL> exec dbms_fga.drop_policy('HR','X_EMP','FGA_XEMP');
```

```
SQL> drop table hr.x_emp purge;
```

Q1. 데이터베이스에 기본으로 설정된 AUDIT_TRAIL의 값이 무엇입니까?

Q2. 객체에 대한 감사옵션을 상세하게 지정하여 감사 기록을 조정할 수 있는 감사패키지의 이름은 무엇입니까?